#29
8. Bond
3/27/03

Due Date: February 3, 2003

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Inventor: Mark E. Davis et al.

Serial #: 08/799,073

Filed: February 11, 1997

Title: STREAMING COMPUTER SYSTEM AND METHOD WITH MULTI-VERSION PROTOCOL COMPATIBILITY

Examiner: Thong H. Vo

Group Art Unit: 2142

Appeal No.: ________________

BRIEF OF APPELLANTS

Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

In accordance with 37 CFR §1.192, Appellants hereby submit the Appellants' Brief on Appeal from the final rejection in the above-identified application, in triplicate, as set forth in the Office Action dated September 6, 2002.

Please charge the amount of $320 to cover the required fee for filing this Appeal Brief as set forth under 37 CFR §1.17(c) to Deposit Account No. 09-0460 of The IBM Corporation, the assignee of the present application. Also, please charge any additional fees or credit any overpayments to Deposit Account No. 09-0460.

I. REAL PARTY IN INTEREST

The real party in interest is the IBM Corporation, the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences for the above-referenced patent application.

III. STATUS OF CLAIMS

Claims 1-8, 10-21, and 23-32 are pending in the application.

On October 15, 1998, a first Office Action was mailed. The first Office Action rejected claims 1-6, 11, 12, and 13-30 under 35 U.S.C. § 102 as being anticipated by United States Patent No. 5,598,276, issued to Cookson et al. Claims 7-8 were rejected under 35 U.S.C. § 103 as being unpatentable over Cookson and further in view of United States Patent No. 5,767,894, issued to Fuller et al. Claims 9-10 were rejected under 35 U.S.C. § 103 as being unpatentable over Cookson in view of Fuller et al. and further in view of Fielding et al.

On January 15, 1999, the Applicants filed Remarks in response to these rejections, leaving the claims unamended.

On March 29, 1999, a Final Office Action was mailed, maintaining the rejections of the first Office Action.

On May 28, 1999, the Applicants filed an Amendment under 37 C.F.R. § 1.116, amending claims 1, 13, 19, 24, 28, 29, and 30.

On June 22, 1999, an Advisory Action was mailed, refusing to enter the amendments because they would require an extended search.

On June 28, 1999, the Applicants filed a Continued Prosecution Application requesting that the unentered amendments be entered.

On August 30, 1999, a first Office Action was mailed. The first Office Action rejected claims 1-7, and 12 under 35 U.S.C. § 102(a)(e) as unpatentable over U.S. Patent 5,790,802, issued to Van Loom et al. Claims 8-11 were rejected under 35 U.S.C. § 103 as unpatentable over Van Loom in view of U.S. Patent No. 5,893,908, issued to Cullen et al. Claims 13-30 were rejected under analogous rationale to the rejections of claims 1-12.

On November 24, 1999, the Applicants filed an Amendment canceling claims 9 and 22, amending claims 1, 10, 19, and 29, and adding claims 31 and 32.

On February 18, 2000, a second Office Action was issued. The second Office Action rejected claims 1-8, 10-21, and 23-32 under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 5,790,802, issued to Heath et al. in view of U.S. Patent No. 5,959,543, issued to LaPorta et al. Claims 31 and 32 were indicated as allowable.

On May 17, 2000, the Applicants filed a communication leaving the claims unamended and arguing that the claims are patentable over the references cited in the second Office Action mailed February 18, 2000.

On August 15, 2000, a third Office Action was mailed. The third Office Action rejected claims 1-8, 10-21, and 23-32 as unpatentable over the earlier cited Van Loom reference in view of U.S. Patent No. 4,912,637, issued to Sheedy et al.

On November 15, 2000, the Applicants filed an Amendment amending claims 1 and 13 and arguing the claims are allowable over the cited references.

On December 21, 2000, a Final Office Action was mailed, maintaining the rejection of the third Office Action mailed August 15, 2000.

On March 21, 2001, the Applicants filed a Notice of Appeal.

On May 21, 2001, the Applicants filed an Appellant's Brief.

On August 28, 2001, the Final Office Action was vacated. The Final Office Action indicated that claims 2-8 and 28 were allowable, but that claims 1, 10-21, 23-27, and 29-32 were rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 5,953,506, issued to Kalra (hereinafter referred to as the Kalra reference) in view of U.S. Patent No. 6,029,200, issued to Beckerman (hereinafter referred to as the Beckerman reference).

On November 28, 2001, the Applicants filed Remarks under 37 C.F.R. § 1.111. The claims remained unamended.

On March 27, 2002, a non-Final Office Action was mailed. The non-Final Office Action (which erroneously indicates that it is in response to Applicants' Request to Reconsider filed January 11, 2002) indicated that claims 2-8 and 28 were allowable, but rejected claims 1, 10-21, 23-27, and 29-32 under 35 U.S.C. 102(e) as being anticipated by the Kalra reference (the previous rejection was under 35 U.S.C. § 103).

On June 26, 2002, the Applicants filed Remarks (erroneously indicated as an Amendment) under 37 C.F.R. § 1.111.

On September 6, 2002, a Final Office Action was mailed, again allowing claims 2-8 and 28, while rejecting claims 1, 10-21, 23-27 and 29-32 under 35 U.S.C. § 102(e) over the Kalra reference.

On November 6, 2002, the Applicants filed Remarks under 37 C.F.R. 1.116.

On November 18, 2002, an Advisory Action was mailed, refusing to enter proposed amendments (although no amendments were proposed in the 1.116 Amendment), and maintaining the rejections of the Final Office Action.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been made subsequent to the final Office Action.

V. SUMMARY OF THE INVENTION

The Applicants' invention is described by a method, apparatus for transmitting a data segment 26 (FIG. 2) in a data stream (FIG. 2) using a write module 22 (FIG. 2) which implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version (specification, page 9, lines 8-16). The method comprising the steps of: (a) outputting a first stream of data according to a first version of the streaming protocol; (b) sequentially appending additional streams of data to the first stream of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and (c) delimiting the data segment in the data stream using begin and end tags (described in FIGs. 3 and 4 and text appurtenant thereto).

VI. ISSUES PRESENTED FOR REVIEW

Whether claims 1, 10-21, 23-27 and 29-32 are unpatentable under 35 U.S.C. § 102(e) over the Kalra reference.

VII. GROUPING OF CLAIMS

The rejected claims do not stand or fall together. Each claim is independently patentable. Separate arguments for the patentability of each claim are provided below.

VIII. ARGUMENTS

A. The Independent Claims 1, 19, and 29 are Patentable Over The Prior Art

1. The Kalra Reference

U.S. Patent No. 5,953,506, issued September 14, 1999 to Kalra et al. discloses a method and apparatus that provides a scalable media delivery system. The present invention provides an apparatus and method for encoding, storing, transmitting and decoding multimedia information in the form of scalable, streamed digital data. A base stream containing basic informational content and subsequent streams containing additive informational content are initially created from standard digital multimedia data by a transcoder. Client computers, each of which may have different configurations and capabilities are capable of accessing a stream server that contains the scalable streamed digital data. Each different client computer, therefore, may access different stream combinations according to a profile associated with each different client computer. Thus, the streams accessed from the server are tailored to match the profile of each client computer so that the best combination of streams can be provided to maximize the resolution of the 3D, audio and video components.

2. Independent Claims 1, 13, 19, 24, 28, 29, and 30 are Patentable Over the Kalra Reference

With Respect to Claims 1, 19, and 29: Claim 1 recites:

> *(a) outputting a first stream of data according to a first version of the streaming protocol;*
> *(b) sequentially appending additional streams of data to the first stream of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and*
> *(c) delimiting the data segment in the data stream using begin and end tags.*

According to the Final Office Action, Kalra teaches a method of transmitting a data segment in a stream using a write module of the type which implements one of a plurality of versions of a

streaming protocol [and] outputting a first stream of data according to a first version of the streaming protocol as described below:

> *It is, therefore, an object of the present invention to provide a method and apparatus for reproducing sounds and/or images with a resolution that is optimized to the capabilities of the client computer that is decoding previously encoded sounds and/or images.*
>
> *It is also an object of the present invention to provide a method and apparatus for encoding digital data representing sounds and/or images as base streams and additive streams of digital data.*
>
> *It is another object of the present invention to provide a method and apparatus for transmitting base streams and a desired number of additive streams of digital data from a stream server to a client computer based on a profile obtained from the client computer.*
>
> *It is a further object of the present invention to provide a method and apparatus for decoding base streams and additive streams of digital data to allow for accurate reproduction of sounds and images.*
>
> *It is a further object of the present invention to provide a method and apparatus that allows for variation in resolution of different media forms so that the quality of a media form such as sound can be increased at the expense of the quality of another media form, such as picture image, according to the desires of the user.*
>
> *It is a further object of the present invention to provide a method and apparatus that allows minimal processing by the server to achieve the objects recited above.*
>
> *In order to obtain the objects recited above, among others, the present invention provides an apparatus and method for encoding, storing, transmitting and decoding multimedia information in the form of scalable, streamed digital data. A base stream containing basic informational content and subsequent streams containing additive informational content are initially created from standard digital multimedia data by a transcoder. Client computers, each of which may have different configurations and capabilities are capable of accessing a stream server that contains the scalable streamed digital data. Each different client computer, therefore, may access different stream combinations according to a profile associated with each different client computer. Thus, the streams accessed from the server are tailored to match the profile of each client computer so that the best combination of streams can be provided to maximize the resolution of the 3D, audio and video components. Since different stream combinations can be accessed, this advantageously allows for the various combinations of content and resolution that are tailored to match that of the specific client computer. If desired, however, the profile can be further adapted to increase the resolution of certain characteristics, such as sound, at the expense of other characteristics, such as video. (col. 1, line 66 - col. 2 line 50)*

The Final Office Action also indicates that Kalra teaches sequentially appending additional data streams of data to the first data stream according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol as follows:

> *The present invention provides an apparatus and method for encoding, storing, transmitting and decoding multimedia information in the form of scalable, streamed digital data. A base stream containing basic informational content and subsequent streams containing additive informational content are initially created from standard digital multimedia data by a transcoder. Client computers, each of which may have different configurations and capabilities are capable of accessing a stream server that contains the scalable streamed digital data. Each different client computer, therefore, may access different stream combinations according to a profile associated with each different client computer. Thus, the streams accessed from the server are tailored to match the profile of each client computer so that the best combination of streams can be provided to maximize the resolution of the 3D, audio and video components.* (Abstract)

It has been found that the present invention can be most easily implemented if a virtual channel for each different type of multimedia is generated. Thus, if only audio and video is being transmitted, two virtual channels, having bandwidth split between them, are needed. However, if audio, video and 3D are all being transmitted, three virtual channels, having bandwidth split between them, are needed. Such virtual channels allows for independent operation of encoders and adaptive stream processors as described hereinafter with respect to the adaptive servers, as well as independent operation of decoders on the client computer. Synchronization can take place through the use of a master clock or be based upon using an audio signal as a master clock. (col. 4, lines 33-46)

... sequence start code, in step 182 there is next searched for the MPEG picture start code, since the codes prior to that are not needed for generation of the .SIGMA.1-.SIGMA.7 additive adaptive streams. Thereafter, in step 184, an adaptive stream picture start code, which corresponds to that specific additive adaptive stream (one of $\sum 1$-$\sum 7$) is written. At that time, a temporal reference that identifies which picture in the group that this particular picture corresponds to is also written. Step 185 follows and a memory allocation for adaptive stream picture header information is made. With reference to FIG. 7C, this information is identified as information 154A, more specifically the next picture pointer and drop frame code. Further explanation of how the next picture pointer and drop frame code are obtained and inserted into this allocated memory will be described hereinafter with reference to FIG. 9C (col. 10, lines 1-17)

The Kalra reference teaches sequentially appending additive data streams (ostensibly for additional resolution) of the same protocol into a single segment. The system described in Kalra can be used with different formats (e.g. MPEG and WAV), but it cannot be used with different protocols within the same data segment.

Kalra teaches transmitting a data stream with additive components of increasing resolution. Kalra does not address, nor does it permit the use of different protocols within data segments. And why should it? The focus of the Kalra reference is transmitting data of differing resolution to computer systems that can utilize the additional resolution. Nowhere does the Kalra reference even remotely suggest that the disclosed method be used with different protocol versions. In fact, the protocol used in the preferred embodiment of the Kalra reference (MPEG encoding) is quite detailed in application and utterly incompatible with non MPEG-compliant coding techniques.

The Final Office Action responded to the foregoing by noting that the claim language did not teach (does not recite?) different protocols. Plainly, claim 1 recites different protocol versions.

The Advisory Action further responded to the foregoing that noting that the "prior art taught different adaptive streams which is obvious each stream represents to [sic] a different version [Kalra col 5 lines 24-55] or different resolution stream [Kalra col. 26 line 49-col 27 line 14]."

The cited portion of the Kalra reference is presented below:

"With respect to the audio sequence 27, different adaptive audio streams are created, with mono being a base channel, and stereo and quadraphonic channels being additive. Further, sounds can be oversampled to even further subdivide such audio streams."

Each media stream in the adaptive stream system according to the present invention is individually scalable as has been previously described. Thus, an application can modify the content it receives from the server as well as what part of this content it has to process to match the bandwidth and computational resources available to it. In addition to these constraints, when a video is embedded in a 3D world, its image on the screen changes considerably depending on where the object on which this video is mapped is relative to the simulated camera. Consider FIG. 28. Videos 901 and 902 are textured on their respective objects. When the objects are mapped onto the screen 906 using the camera 905, the image of video 901 is the thick line 903 and the image of video 902 is the thick line 904. Image 903 is much smaller than image 904. This projection process is essentially limiting the information that ends up being displayed on the screen. This fact can be used to reduce the computational and bandwidth resources, by sending a different resolution stream to video 901 as compared to video 902. As the camera and/or object moves around in the scene, this resolution of the video can be changed continuously. In the present media architecture, this 3D information will be changed into a user-driven profiles to control the information content in each of the videos 901 and 902 as explained later. Typically the different videos in a 3D scene will be at different distances from the camera and a number of videos can be simultaneously displayed using this technique. If multiple videos were to be displayed without this 3D driven control of video content, one would have to decode each of the videos at full resolution and then decimate them to map to the screen to the proper size. This would involve two resource wasting operations, full decode and decimate which is avoided in this implementation.

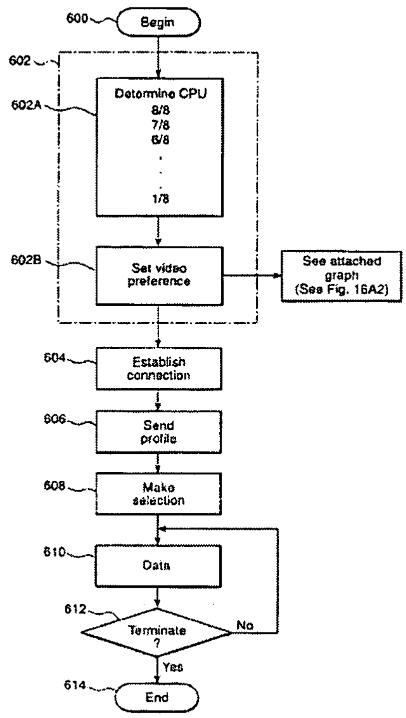
The foregoing clearly does not refer to different protocol versions or additive protocol versions.

Finally, the Office Action indicates that the Kalra reference teaches delimiting the data segment in the data stream using begin and end tags as follows:

The stream management step 814 using a stream management module is the outgoing interface to the server that sends the stream modification messages determined above to the server. Packetized commands are sent to the server to among other things, STOP or RESUME data associated with a particular object identification, change PRIORITY of the specified type of data for the specified type of object, STOP data for all objects associated with a particular data identification, or START data for all objects associated with a particular data identification. (col. 25, lines 39-47).

and, in FIG. 16A1, reproduced below:

Fig. 16A1



The foregoing text discloses the transmission of packetized commands ... not begin and end tags. The Applicants do not understand how FIG. 16A1 is relevant at all.

In the Advisory Action, the Examiner responded that "the prior art taught a single video adaptive with header information, begin (i.e. start code) and end signals [Kalra col 17 lines 55-col 19 lines 22]." This passage is disclosed below:

> *Step 3: Video Preference Constraint*
>
> *The profile indicates the video preference set for best spatial resolution (8/8). This selects the single video adaptive stream indicated in FIG. 15B2D.*
>
> *Once step 552 in FIG. 15A is completed and the stream combination is set, the transmitting of streams by the server, and the reception of the same by the client computer then takes place.*
>
> *With reference to FIG. 15C, the transmission sequence begins with, in step 554A, the sending of an adaptive stream identification and header information, in which the codes indicating the specific adaptive streams that will be sent and other MPE G and adaptive stream header information as has been previously specified. In step 554B that follows, the group codes and headers, are transmitted, and, thereafter in step 554C, the picture code is transmitted. For each picture, in step 554D1 the complete 3DZO sequence is transmitted and in step 554D2 the .SIGMA.1 through .SIGMA.7 additive adaptive streams are transmitted, as determined by*

the profile, as has been discussed previously.

The drop frame codes and next picture pointer need not be transmitted, as these codes are used by the stream server to quickly determine whether to drop a frame and then determine quickly the location of the next frame, so that a real-time, appropriate, and dynamically changing picture sequence, dependent upon the profile, can be transmitted. This transmission occurs for each picture in a group, and, then each group of pictures, until transmission of the entire sequence takes place. Although it should be apparent, it is noted that the streams that need to be transmitted from the server can be quickly determined by the server processor, since the server processor can use the next picture pointer and drop frame codes embedded in the data structure to quickly determine which frames to send, as well as which frames not to send, depending on the particular profile.

In an alternate implantation of the data structure illustrated in FIGS. 7A and 7B, there can be created a set of two files, an index file and a data file. In the data file is stored the start codes, header data, and actual video data associated with each of the adaptive streams as has been previously described. In the index file is stored drop frame codes for each adaptive stream, down to the slice level, as well as pointers to the location for each slice of the data for the appropriate data that will be transmitted if a frame is not dropped. Using this data file structure, the processor can determine even more quickly whether a particular frame, and which adaptive streams within the frame, should be transmitted.

At the end of a group code sequence, whether a profile update has occurred is checked in step 554E. If a profile update has occurred, then step 550 of FIG. 15A follows and a new profile is received. If there is not a new profile, then step 554B follows and a new group code, and corresponding pictures, each with corresponding adaptive streams is transmitted, which operation continues until the end of a sequence.

On the client computer reception side, step 610 of FIG. 16A1 is further illustrated in FIG. 16B. This reception begins in step 620, in which the adaptive stream and header information transmitted in step 554A of FIG. 15C is received. Steps 622 follows, in which the group code and header information transmitted in step 554B is received. Step 624 receives picture code and picture header information transmitted in step 554C, and, thereafter, in steps 626 and 628 the transmitted .SIGMA.0 sequence and, as determined by the profile, appropriate .SIGMA.1 through .SIGMA.7 additive adaptive streams are received, respectively. Once the data for an entire group of adaptive stream pictures is received, it is then operated upon by an adaptive stream decoder in step 628. Once decoded, this group, which will be a sequence of reconstructed I, B and P pictures, is then operated upon using a standard MPEG decoder in step 630 to obtain reconstructed frames.

If, after a group of pictures is received it is detected that a new profile is desired or is sent, step 602 in FIG. 16A1 follows and a new profile is made. Otherwise, step 622 repeats.

FIG. 16C illustrates operation of the adaptive stream decoder in further detail. As illustrated, in step 650, the group start code and MPEG headers are received. Thereafter, in step 652 the picture start code is received. In step 652 and 654 the picture start code and mpeg picture headers are received, followed, in step 656, with receipt of the slice start code for a particular picture. In steps 658 the MPEG header information is received. Subsequently, in step 660, all of the information corresponding to the adaptive streams for a particular slice is received and blocks of reconstructed DCT coefficients are obtained for those blocks that have DCT coefficients, according to the number of additive adaptive streams that were transmitted. The adaptive stream decoder, having been informed of which additive adaptive streams are being transmitted, as well as the number of frames per second and other needed synchronizing information, is capable of reconstructing the DCT coefficient matrix for each block. Thereafter, in step 662, the write correction code, if any, is received and used to correct the drift introduced in the client decoder because of the reduced transmission stream (i.e less than all of the additive adaptive streams).

The Applicants respectfully disagree that anything in the foregoing fairly teaches the use of beginning and end tags to delimit data segments having sequentially appended streaming protocol versions.

For all of the foregoing reasons, the Applicants respectfully traverse the rejection of claim 1.

Claims 19 and 29 include limitations analogous to those of claim 1 and are patentable on the same basis.

With Respect to Claims 13, 24, and 30: According to the Office Action, claim 13 included limitations similar to those of claim 1, except that it includes the step of testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the addition[al] stream of data according to the selected version as an "inherent" feature in encoding, decoding, storing, and transmitting a data stream. Although the Office Action expressly indicated it is relying on the inherency doctrine, the Office Action then proceeded to recite 10 different sections of text which purport to disclose these features.

The Applicants reviewed the cited portions of the Kalra reference, and indicated that they could not determine where the testing step was disclosed. The Applicants therefore concluded that the Examiner is relying on the inherency doctrine in rejecting these claims, and traversed.

Inherency "may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." *Continental Can Co. v Monsanto Co.*, 948 F.2d 1264, 1269(Fed. Cir. 1991). Instead, to establish inherency, the extrinsic evidence "must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill." *Continental Can Co.*, 948 F.2d at 1268. In finding anticipation by inherency, the Office Action ignored the foregoing critical principles. The Office Action did not show that the "testing" must be performed (in fact, it need not be ... another reference cited by the Examiner in a previous Office Action used a "brute force" approach wherein the length of the data is separately transmitted, obviating the need for any such testing) or that the "testing" be performed prior to receiving each additional stream of data.

In the Final Office Action, another section (col. 25, lines 38-48) of the Kalra is offered as disclosing the step of *testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the additional stream of data according to the selected version.* It is shown as follows:

Thus, as shown in FIG. 26, based on previous frame statistics, it is determined as to what the new priorities of different components in the scene should be, whether to add or remove vertices, change rendering modes such as flat shaded, gouraud shaded, phong shaded, gouraud lighting model, phong lighting model, texturing enable disable, and resolution of texturing, or increase or decrease viewport size, i.e. the size of the window in which the frame is rendered. Thus, a determination can be made whether and how to render each different visible object and, therefore, what data will be needed for the next frame that will be rendered. Based upon the level of detail evaluated, two actions result. One, control messages to be sent to the server are determined that modify the relative rate of data transmission, both overall as well as for each object. Second data from the data dictionary 802 is merged into the current frame data buffer 804 so that the next frame can be rendered.

The stream management step 814 using a stream management module is the outgoing interface to the server that sends the stream modification messages determined above to the server. Packetized commands are sent to the server to among other things, STOP or RESUME data associated with a particular object identification, change PRIORITY of the specified type of data for the specified type of object, STOP data for all objects associated with a particular data identification, or START data for all objects associated with a particular data identification.

Other than by the use of impermissible hindsight reconstruction, the Applicants are at a loss to explain how the foregoing can be interpreted to fairly teach the above features recited in claim 13.

Claims 24 and 30 are patentable for the same reasons.

B. The Dependent Claims Are Patentable Over The Prior Art

1. Dependent Claims 10-12, 14-18, 20-21, 23, 25-27 and 31-32 are Patentable Over the Kalra Reference(s)

<u>With Respect to Claim 10</u>: Claim 10 includes the limitations of claim 1 and is patentable on this basis. The Office Action indicates that this limitation can be found in FIG. 16A1 of the Kalra reference (reproduced above), but the Applicants cannot ascertain how FIG. 16A1 can be interpreted as such.

<u>With Respect to Claims 11 and 23</u>: Claim 11 recites:

> *determining whether the data segment is stored in a current context for the data stream;*
> *if so, transmitting an alias tag in lieu of the data segment; and*
> *if not, storing the data segment in the current context.*

According to the Office Action, these features are disclosed in the Kalra reference in FIG. 16A1 and as recited below:

The stream management step 814 using a stream management module is the outgoing interface to the server that sends the stream modification messages determined above to the server. Packetized commands are sent to the server to among other things, STOP or RESUME data associated with a particular object identification, change PRIORITY of the specified type of data for the specified type of object, STOP data for all objects associated with a particular data identification, or START data for all objects associated with a particular data identification. (col. 25, lines 39-47).

The Applicants frankly do not understand how the foregoing text or FIG. 16A1 can be interpreted to disclose the features of claims 11 and 23. Accordingly, the Applicants traverse this rejection.

<u>With Respect to Claim 12</u>: Claim 12 includes the limitations of claim 1, and is patentable on this basis.

<u>With Respect to Claims 14 and 25</u>: Claim 14 recites:

if the end of the data segment has not been detected upon receiving the additional stream of data according to the selected version, disregarding any remaining data in the data segment.

According to the Office Action, this feature is disclosed as follows:

Overall operation of the adaptive stream server will now be described with respect to FIG. 15A. Once the adaptive stream server receives a profile from the user, in step 550, it uses that information, as well as other information described hereinafter, to make a determination of which streams to transmit in a step 552. Once this determination is made, streams are actually transmitted in a step 554, as long as the profile is not updated, as will be explained further hereinafter, or there is no indication that there is an end of session, as depicted in FIG. 15A by step 556, transmission will continue. If an end of session is depicted, the end of the session will occur as indicated by step 568.

The foregoing indicates what happens if an end of session is detected, not end of data. Further, there is no teaching to disregard data segments if the end of the data segment has not been detected upon receiving the additional stream of data.

For the foregoing reasons, the Applicants respectfully traverse the rejection of claim 14.

Claim 25 recites features analogous to those of claim 14 and is patentable for the same reasons.

<u>With Respect to Claim 15</u>: According to the Office Action, the Kalra reference discloses storing the data segment in a current context, including any disregarded data in a number of locations in the specification. The Applicants have reviewed these passages and cannot determine where such disclosure might be found. Further, claim 15 includes the limitations of claim 14 and 13 and is patentable on this basis alone.

With Respect to Claims 16, 20, and 26: Claims 16, 20, and 26 include the limitations of claim 13 and are patentable on this basis.

With Respect to Claims 17 and 21: In rejecting claims 17 and 21, the Office Action improperly relies on the inherency doctrine. Accordingly, the Applicants traverse this rejection.

With Respect to Claim 18: Claim 18 recites:

receiving an object type for the data segment; and
allocating and initializing an object based upon the object type to build the object from the streams of data in the data segment.

According to the Office Action, these features are disclosed in FIG. 22 , which is reproduced below:

Object ID	Type	Data Pointer	Priority
100	Geometry	xABCD04	...
110	Texture	x34598C	
130	Material	xABC234	

While the foregoing discloses the use of object IDs, types, and data pointers to objects, the Applicants do not see where the foregoing discloses allocating and initializing an object based on the object type. Accordingly, the Applicants traverse this rejection.

With Respect to Claims 31 and 32: The Office Action indicates that the limitations of claims 31 and 32 can be found in FIG. 16A1 of the Kalra reference. The Applicants respectfully disagree, and traverse this rejection as well.

IX. CONCLUSION

In light of the above arguments, Appellants respectfully submit that the cited references do not anticipate nor render obvious the claimed invention. More specifically, Appellants' claims recite novel physical features which patentably distinguish over any and all references under 35 U.S.C. §§ 102 and 103. As a result, a decision by the Board of Patent Appeals and Interferences reversing the Examiner and directing allowance of the pending claims in the subject application is respectfully solicited.

Respectfully submitted,

GATES & COOPER LLP

Attorneys for Applicants

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: February 3, 2003

By: Victor G. Cooper
Name: Victor G. Cooper

G&C 30571.77-US-01

APPENDIX

1. A method of transmitting a data segment in a data stream using a write module which implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, the method comprising the steps of:

(a) outputting a first stream of data according to a first version of the streaming protocol;

(b) sequentially appending additional streams of data to the first stream of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and

(c) delimiting the data segment in the data stream using begin and end tags.

2. The method of claim 1, further comprising the step of receiving the data segment from a data stream using a read module of the type which implements a second selected one of the plurality of versions of the streaming protocol, the receiving step including the steps of:

receiving the first stream of data;

if the second selected version is earlier than the first selected version, receiving each additional stream of data according to each subsequent version of the streaming protocol up to and including the second selected version, and disregarding any remaining data in the data segment;

if the second selected version is equal to or later than the first selected version, sequentially receiving the additional streams of data according to each subsequent version of the streaming protocol up to and including the second selected version; and

testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the additional stream of data according to the second selected version.

3. The method of claim 2, wherein the data segment is an object.

4. The method of claim 3, wherein the data segment includes all of the data necessary to reconstruct the object; whereby the data stream is serial.

5. The method of claim 3, wherein the testing step includes the step of initializing object data that is not received from the data stream to a default value.

6. The method of claim 3, further comprising the steps of:

transmitting an object type for the data segment; and

receiving the object type, including allocating and initializing an object when receiving the data segment based upon the object type.

7. The method of claim 2, wherein the read and write modules are resident on the same computer.

8. The method of claim 2, wherein the read and write modules are resident on separate computers.

9. (CANCELED)

10. The method of claim 1, wherein no additional tags are embedded in the data segment between the begin and end tags.

11. The method of claim 1, further comprising the steps of:

determining whether the data segment is stored in a current context for the data stream;

if so, transmitting an alias tag in lieu of the data segment; and

if not, storing the data segment in the current context.

12. The method of claim 1, wherein the data stream is a non-random access data stream.

13. A method of receiving a data segment from a data stream using a read module which implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, the method comprising the steps of:

(a) receiving a first stream of data according to a first version of the streaming protocol;

(b) if the selected version of the streaming protocol is not the first version of the streaming protocol, sequentially receiving additional streams of data according to each subsequent version of the streaming protocol up to and including the selected version; and

(c) testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the additional stream of data according to the selected version.

14. The method of claim 13, further comprising the step of, if the end of the data segment has not been detected upon receiving the additional stream of data according to the selected version, disregarding any remaining data in the data segment.

15. The method of claim 14, further comprising the step of storing the data segment in a current context, including any disregarded data therefrom.

16. The method of claim 13, wherein the data segment is an object.

17. The method of claim 16, wherein the testing step includes the step of initializing object data that is not received from the data stream to a default value.

18. The method of claim 16, further comprising the steps of:

receiving an object type for the data segment; and

allocating and initializing an object based upon the object type to build the object from the streams of data in the data segment.

19. A computer system that transmits data segment in a data stream, the computer system comprising a write module that implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, and that outputs the data segment in the data stream, wherein:

(a) the write module comprising means for outputting a first stream of data according to a first version of the streaming protocol;

(b) the write module comprising means for sequentially appending additional streams of data to the first stream of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and

(c) the write module comprising means for delimiting the data segment in the data stream using begin and end tags.

20. The computer system of claim 19, wherein the data segment is an object.

21. The computer system of claim 19, wherein the write module further comprises means for transmitting an object type for the data segment.

22. (CANCELED)

23. The computer system of claim 19, wherein the write module further comprises means for transmitting an alias tag in lieu of the data segment if the data segment is stored in a current context for the data stream.

24. A computer system that receives a data segment from a data stream, the computer system comprising a read module that implements a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, and that receives the data segment from the data stream, wherein the read module comprises:

(a) means for receiving a first stream of data according to a first version of the streaming protocol;

(b) means for sequentially receiving additional streams of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and

(c) means for testing whether an end of the data segment has been detected, and if so, for terminating reception of the data segment prior to receiving the additional stream of data according to the selected version prior to receiving each additional stream of data.

25. The computer system of claim 24, wherein, if the end of the data segment has not been detected upon receiving the additional stream of data according to the selected version, the read module disregards any remaining data in the data segment.

26. The computer system of claim 24, wherein the data segment is an object.

27. The computer system of claim 26, wherein the read module comprises means for receiving an object type for the data segment and for allocating and initializing an object based upon the object type to build the object from the streams of data in the data segment.

28. A computer system comprising first and second computers that transmit a data segment in a data stream from the first computer to the second computer, the first computer comprising means for implementing a first selected one of a plurality of versions of a streaming protocol, and the second computer comprising means for implementing a second selected one of the plurality of versions of the streaming protocol, wherein the second selected one of the plurality of versions of the streaming protocol is additive to the first selected on of the plurality of versions of the streaming protocol, and wherein:

(a) the first computer includes a write module for transmitting the data segment, wherein the write module outputs a first stream of data according to a first version of the streaming protocol, and if the first selected version is not the first version of the streaming protocol, the write module sequentially appends to the first stream of data additional streams of data according to each subsequent version of the streaming protocol up to and including the first selected version; and

(b) the second computer includes a read module for receiving the data segment from the first computer, wherein the read module receives the first stream of data, wherein if the second selected version is earlier than the first selected version, the read module receives each additional stream of data according to each subsequent version of the streaming protocol up to and including the second selected version, and disregards any remaining data in the data segment, wherein if the second selected version is equal to or later than the first selected version, the read module sequentially receives the additional streams of data according to each subsequent version of the streaming protocol up to and including the second selected version, and wherein, prior to receiving each additional stream of data, the read module detects whether an end of the data segment has been detected, and if so, terminates reception of the data segment prior to receiving the additional stream of data according to the second selected version.

29. A program storage device, readable by a computer system and tangibly embodying one or more programs of instructions executable by the computer system to perform method steps of transmitting a data segment in a data stream in a format based upon a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, the method comprising the steps of:

(a) outputting a first stream of data according to a first version of the streaming protocol;

(b) sequentially appending additional streams of data to the first stream of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and

(c) delimiting the data segment in the data stream using begin and end tags.

30. A program storage device, readable by a computer system and tangibly embodying one or more programs of instructions executable by the computer system to perform method steps of receiving a data segment from a data stream according to a selected one of a plurality of versions of a streaming protocol wherein each subsequent version of the streaming protocol is additive to a previous version, the method comprising the steps of:

(a) receiving a first stream of data according to a first version of the streaming protocol;

(b) sequentially receiving additional streams of data according to each subsequent version of the streaming protocol up to and including the selected version, if the selected version of the streaming protocol is not the first version of the streaming protocol; and

(c) testing, prior to receiving each additional stream of data, whether an end of the data segment has been detected, and if so, terminating reception of the data segment prior to receiving the additional stream of data according to the selected version.

31. The method of claim 13, wherein the step of testing whether and end of the data segment has been detected comprises the step of testing for a premature end tag and terminating the reception of the data segment when a premature end tag is received.